

Claim Amendment Summary**Claims pending**

- At time of the Action: Claims 1-2 and 10-60.
- After this Response: Claims 1-2 and 10-60.

Cancelled claims: none.**Amended claims:** 51.**New claims:** none.

1. **(PREVIOUSLY PRESENTED)** A method of parsing an Extensible Markup Language (XML) data stream comprising:

defining a plurality of states, individual states being associated with individual elements of an XML data stream;

associating one or more rules with each state;

receiving an XML data stream;

evaluating the XML data stream against one or more of the rules for individual elements contained in the XML data stream; and

processing only those portions of the XML data stream that do not violate any of the rules that are associated with those portions.

2. **(ORIGINAL)** The method of claim 1, wherein the one or more rules relate to a schema of the XML data stream.

3-9. **(CANCELLED)**

1 10. **(ORIGINAL)** The method of claim 1 further comprising defining
2 one or more rules that relate to an element's contents.

3
4 11. **(ORIGINAL)** The method of claim 10, wherein said one or more
5 rules that relate to an element's contents define which elements can be contained
6 within other elements.

7
8 12. **(ORIGINAL)** The method of claim 11, wherein if a rule that defines
9 which elements can be contained within other elements is violated, disregarding
10 associated portions of the XML data stream until a close tag is received for an
11 element that violates the rule.

12
13 13. **(ORIGINAL)** A computer-readable medium having a program
14 thereon which, when executed by a computer, performs the steps of claim 1.

15
16 14. **(PREVIOUSLY PRESENTED)** A method of parsing an Extensible
17 Markup Language (XML) data stream comprising:

18 defining a schema module that is associated with an HTTP request type that
19 is received from a client, the schema module having a function that determines
20 whether an XML data stream conforms to a given schema that is associated with
21 the HTTP request type;

22 evaluating an XML data stream with the schema module; and

23 processing only those portions of the XML data stream that conform to the
24 given schema.
25

1 15. **(ORIGINAL)** The method of claim 14, wherein said defining of the
2 schema module comprises defining a plurality of schema modules, individual
3 schema modules being associated with different HTTP request types.

4
5 16. **(ORIGINAL)** The method of claim 14, wherein said function
6 determines whether there are any unauthorized elements that appear in a client's
7 request.

8
9 17 **(ORIGINAL)** The method of claim 14, wherein said function
10 determines whether there are any unauthorized elements that appear in a client's
11 request; said disregarding comprising disregarding said XML data stream portion
12 until a close tag is received for an unauthorized element.

13
14 18. **(ORIGINAL)** The method of claim 14, wherein said HTTP request
15 type comprises a WebDAV request type.

16
17 19. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
18 request type comprises a PROPFIND request.

19
20 20. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
21 request type comprises a PROPPATCH request.

22
23 21. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
24 request type comprises a SEARCH request.

25

1 22. **(ORIGINAL)** The method of claim 18, wherein said WebDAV
2 request type comprises one of a LOCK and UNLOCK request.

3
4 23. **(ORIGINAL)** A computer-readable medium having a program
5 thereon which, when executed by a computer, performs the steps of claim 14.

6
7 24. **(ORIGINAL)** An Extensible Markup Language (XML) parsing
8 system comprising:

9 a parser configured to receive an XML data stream and generate a series of
10 calls as it parses the XML data stream;

11 a node factory communicatively associated with the parser and configured
12 to receive the parser's calls and responsive thereto construct a representation of the
13 XML data stream that the parser is parsing; and

14 a schema module communicatively associated with the node factory and
15 configured to evaluate the node factory's representation of the XML data stream
16 and determine whether it conforms to a known schema.

17
18 25. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing
19 system comprises a plurality of schema modules, each schema module being
20 associated with a different known schema.

21
22 26. **(ORIGINAL)** The parsing system of claim 24, wherein the schema
23 module corresponds to an HTTP request type.

24
25

1 27. **(ORIGINAL)** The parsing system of claim 24, wherein said parsing
2 system comprises a plurality of schema modules, each schema module being
3 associated with a different known schema and corresponding to a different HTTP
4 request type.

5
6 28. **(ORIGINAL)** The parsing system of claim 27, wherein at least one
7 of the different HTTP request types is a WebDAV request.

8
9 29. **(ORIGINAL)** The parsing system of claim 24, wherein the schema
10 module is configured to ignore an XML data stream portion that does not conform
11 to the known schema.

12
13 30. **(ORIGINAL)** An Extensible Markup Language (XML) parsing
14 system comprising:

15 a collection of schema modules, each of which being configured to evaluate
16 a different schema that is associated with an XML data stream; and

17 a plurality of states associated with each schema module, individual states
18 of a schema module defining a schema requirement relating to a particular element
19 that is evaluated by that schema module.

20
21 31. **(ORIGINAL)** The parsing system of claim 30, wherein each
22 schema module is associated with a different HTTP request and is configured to
23 evaluate a schema that is associated with the HTTP request with which is it
24 associated.

25

1 32. **(ORIGINAL)** The parsing system of claim 31, wherein at least one
2 of the HTTP requests is a WebDAV request.

3
4 33. **(ORIGINAL)** The parsing system of claim 31, wherein each of the
5 HTTP requests is a WebDAV request.

6
7 34. **(PREVIOUSLY PRESENTED)** A method of parsing an Extensible
8 Markup Language (XML) data stream comprising:

9 defining a plurality of states, individual states being associated with
10 individual elements of an XML data stream;

11 associating one or more rules with each state;

12 receiving an XML data stream;

13 evaluating the XML data stream against one or more of the rules for
14 individual elements contained in the XML data stream; and

15 disregarding associated portions of the XML data stream if any of the rules
16 that are associated with those portions are violated, the disregarded portions of the
17 XML data stream representing at least one error in the XML data stream.

18
19 35. **(PREVIOUSLY PRESENTED)** The method of claim 34, wherein
20 the request type is a WebDAV request type.

21
22 36. **(PREVIOUSLY PRESENTED)** The method of claim 35, wherein
23 the WebDAV request type is a PROPFIND request.

24
25

1 41. **(PREVIOUSLY PRESENTED)** The method of claim 40, wherein
2 each schema module is associated with at least one request type that defines the
3 XML data stream.

4
5 42. **(PREVIOUSLY PRESENTED)** The method of claim 41, wherein
6 the request type is a WebDAV request type.

7
8 43. **(PREVIOUSLY PRESENTED)** The method of claim 42, wherein
9 the WebDAV request type is a PROPFIND request.

10
11 44. **(PREVIOUSLY PRESENTED)** The method of claim 42, wherein
12 the WebDAV request type is a PROPPATCH request.

13
14 45. **(PREVIOUSLY PRESENTED)** The method of claim 42, wherein
15 the WebDAV request type is a SEARCH request.

16
17 46. **(PREVIOUSLY PRESENTED)** The method of claim 42, wherein
18 the WebDAV request type is one of a LOCK and UNLOCK request.

19
20 47. **(PREVIOUSLY PRESENTED)** The method of claim 40 further
21 comprising defining one or more rules that relate to an element's contents.

22
23 48. **(PREVIOUSLY PRESENTED)** The method of claim 47, wherein
24 said one or more rules that relate to an element's contents define which elements
25 can be contained within other elements.

1
2 49. **(PREVIOUSLY PRESENTED)** The method of claim 48, wherein
3 if a rule that defines which elements can be contained within other elements is
4 violated, disregarding associated portions of the XML data stream until a close tag
5 is received for an element that violates the rule.

6
7 50. **(PREVIOUSLY PRESENTED)** A computer-readable medium
8 having a program thereon which, when executed by a computer, performs the
9 steps of claim 40.

10
11 51. **(CURRENTLY AMENDED)** A method of parsing an Extensible
12 Markup Language (XML) data stream comprising:

13 defining a plurality of states, individual states being associated with
14 individual elements of an XML data stream, wherein the defining of the plurality
15 of states comprises defining one or more schema modules that are configured to
16 track one or more states of the XML data stream, each schema module being
17 associated with at least one request type that defines the XML data stream;

18 associating one or more rules with each state;

19 receiving the XML data stream;

20 evaluating the XML data stream against one or more of the rules for
21 individual elements contained in the XML data stream; and

22 disregarding associated portions of the XML data stream if any of the rules
23 that are associated with those portions are violated.
24
25

1 52. **(PREVIOUSLY PRESENTED)** The method of claim 51, wherein
2 the request type is a WebDAV request type.

3
4 53. **(PREVIOUSLY PRESENTED)** The method of claim 52, wherein
5 the WebDAV request type is a PROPFIND request.

6
7 54. **(PREVIOUSLY PRESENTED)** The method of claim 52, wherein
8 the WebDAV request type is a PROPPATCH request.

9
10 55. **(PREVIOUSLY PRESENTED)** The method of claim 52, wherein
11 the WebDAV request type is a SEARCH request.

12
13 56. **(PREVIOUSLY PRESENTED)** The method of claim 52, wherein
14 the WebDAV request type is one of a LOCK and UNLOCK request.

15
16 57. **(PREVIOUSLY PRESENTED)** The method of claim 51 further
17 comprising defining one or more rules that relate to an element's contents.

18
19 58. **(PREVIOUSLY PRESENTED)** The method of claim 57, wherein
20 said one or more rules that relate to an element's contents define which elements
21 can be contained within other elements.

22
23
24
25

1 59. **(PREVIOUSLY PRESENTED)** The method of claim 58, wherein
2 if a rule that defines which elements can be contained within other elements is
3 violated, disregarding associated portions of the XML data stream until a close tag
4 is received for an element that violates the rule.

5
6 60. **(PREVIOUSLY PRESENTED)** A computer-readable medium
7 having a program thereon which, when executed by a computer, performs the
8 steps of claim 51.
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25